Early Experiences with BG/L

Susan Coghlan
<smc@mcs.anl.gov>
Argonne National Laboratory

MCS BG/L Specs

- 1 Rack (1024 nodes)
- 32 I/O nodes (1/32 IO/Compute ratio)
- 4 Frontends (JS20 blades – PPC970 2.1GHz dual-cpu 4GB RAM) [SLES9]
- 1 Service Node (4-way 1.7 Ghz PPC (2 CPU cores), 16GB RAM) [SLES8]
- 20 Storage servers (4 homedir, 16 PVFS) ~14TB [SLES9]

Time Line

- Install started 1/20/2005
- Linpack ran successfully 1/21/2005
- Acceptance delayed
  - Waiting for storage nodes and frontends
  - Problems with one of our applications
- Machine accepted 1/31/2005
- Users on running apps 2/2/2005

Installation Observations

- BG/L Installation is clearly a WIP
  - We benefited from LLNL and SDSC installations
- Having a process is useful, but
- Generally went well
- The hardware is not the problem

Installation Issues

- Hardware Problems (Minimal)
  - 2 Compute nodes (+1 last week)
  - 4 I/O nodes
- Configuration Quirks, things like:
  - Must define 64 I/P for I/O nodes even though we only have 32
  - Delete/reload block definitions, then double allocate required after boot
- Software Problems
  - everyone needs a FLASH in their acceptance test suite

The Mission

- System Software Development
- Scientific Application Porting
- Performance Testing and Benchmarking
- Community Resource

- Our operation at ANL must be extremely flexible!

Making it Usable

- System Software Modifications
  - Filesystem
  - Resource Manager
  - Partition Management – ANL
  - I/O Node Environment
- Rational User Environment
- System Management Needs

Filesystems

- NFS – make sure your rc.d script(s) retry enough times
- Other Filesystems [LLNL (Lustre), SDSC (GPFS) and ANL (PVFS2)]
- PVFS running as a production filesystem on Jazz for the past 2 years
- Mounted across full BGL rack
- Performance tuning in progress

Resource Management

- LLNL(SLURM), SDSC(LoadLeveler), ANL (no-name)
- Developed at MCS, in use on Chiba
- Opensource components based on the SSS component interfaces
- Lightweight implementations written in python
- BGL Components:
  - Process Manager (process startup and control)
  - Queue Manager
  - Scheduler (currently simple FIFO)
  - Allocation Manager (integration not completed)
- In Alpha Testing

ANL Ramdisk and I/O kernel

- ANL Linux I/O node toolkit available for distribution
- Linux kernel, config, compile & ramdisk tools, etc.
- Open source distribution of the I/O node ramdisk and kernel
- We are currently using it to extend the capabilities of the I/O node, and to build performance tools (TAU) and kernel modules (PVFS)

Building the User Env

- As installed, not robust (csh broken)
- Installed Softenv (developed at MCS)
- Moved all the BGL-isms under Softenv
- Integrated into MCS account management
- Prepared partitions
- Created mail lists – discuss and notify
- Built a status page

Tools for the Users

- Bgl-list
  - List jobs: running, errored, completed
  - List blocks: allocated, all
  - Long listing: all data found (* location)
  - By ID, * by user, * by partition
- * Tool for processing logs, RAS events
- * Tool for cleaning up hung jobs
- * Tool for managing mapping

Current State

- Operating mode:
  - 32 node 'developer' partitions
  - Co-processor and Virtual mode versions
  - Small set with personal ramdisks and kernels
  - Evenings/weekends 512 and 1024 node runs
- 26 active users
- ~3000 jobs run (80% in developer)

A Few of the Projects

- nQMC
- NeoCortex
- Nanocatalysis
- QCD
- POP
- MPIBlast
- TAU/PDT
- PETSc
- MPE/Jumpshot

Early User Experiences (what they liked)

- Code compiled and ran first time
- Fast communications
- Nice scaling
- Ability to map processes to underlying topology is cool (if I could get it to work)
- It's certainly a challenge

Early User Experiences (what they didn't like)

- Lack of useful information
- Lack of documentation
- Compiler is buggy, options don't work as expected and diagnostics are poor
- Lack of a simple mpirun that just works. Options don't work as expected.
- Debugging is Hard!

Support So Far

- Not thrilled with the support interface
- Not thrilled with the response times
- Quick solutions once we made contact
- We haven't pushed hard, yet
- but
- our bug list is growing rapidly
- some are most likely fixed in newer versions

ANL BG/L Community Resources

- ANL BG/L Wiki (available now)
  - http://www.bgl.mcs.anl.gov/wiki
- MCS BG/L Web site (available now)
  - http://www.bgl.mcs.anl.gov
- Problem tracking system (available soon)
  - http://www.bgl.mcs.anl.gov/support

Prioritized List – Users View

- Debugging tools
- Software upgrades – compiler, mpi
- Fully functional compiler with good diagnostics and correct assembly code
- Fully functional mpirun
- Documentation, all sorts, more of.
- Documentation that matches reality
- PAPI, etc.

Users list, continued

- Mapping documentation
- Usable error information
- Correct date/time and timing mechanisms that go beyond 7hr18m (MPI_GET_TIME, Fort DATE_AND_TIME, cpu time)
- Dynamic libraries
- MIMD support

Prioritized List – SysAdmin View

- Quick access to error translations to understand failure modes
- Simplified startup process
- No VNC requirement (VNC – big mistake)
- DB2 – rational protections, documentation for relations, schema, etc.
- Documented interface RAS/etc [for automated monitoring, i.e. nagios]

SysAdmin list, cont.

- Up-paced software updates (with better revision numbering)
- Better support model
- Rational I/O node environment
- Source for ciod,mpi,mmcs_*